In the claims:

1. A lock-step synchronism fault-tolerant computer system including a plurality of computing modules having a processor and a memory in which each computing module processes the same instruction string in synchronization with each other, wherein

when detecting disagreement in a state of access to an external bus among respective said processors in each said computing module, if no fault is detected in the system including each said computing module, synchronization among each said computing module is recovered by adjusting timing of a response to an access which each said processor executes as a synchronization control instruction by an interruption.

2. The fault-tolerant computer system as set forth in claim 1, further comprising:

a fault detector which monitors existence/non-existence of a fault in the entire system;

a bus monitor which monitors an access of the processor in each said computing module to the external bus and when detecting disagreement in output among the respective computing modules, if no fault is detected by said fault detector, notifying an interruption to each said processor, and

a synchronization controller which re-

synchronizes each computing module by adjusting timing of a response to an access from each said processor which is caused by said interruption.

3. The fault-tolerant computer system as set forth in claim 2, wherein

said bus monitor, when detecting disagreement in output among the respective computing modules, if no fault is detected by said fault detector, interrupts each said processor with a predetermined task, which is a task of executing an access to a predetermined resource in said synchronization controller, to re-synchronizing the computing modules, and

said synchronization controller transmits a response to all the computing modules simultaneously, when receiving accesses to said resource from all the processors.

4. The fault-tolerant computer system as set forth in claim 2, wherein

a plurality of pairs of said bus monitor, said fault detector and said synchronization controller are provided.

5. The fault-tolerant computer system as set forth in claim 2, wherein

said bus monitor, said fault detector and said

synchronization controller are provided in a peripheral device control unit which controls a peripheral device and connected to the external bus in said computing module through a PCI bridge.

6. A re-synchronization method in a lock-step synchronism fault-tolerant computer system including a plurality of computing modules having a processor and a memory in which each computing module processes the same instruction string in synchronization with each other, comprising the steps of:

when detecting disagreement in a state of access to an external bus among respective said processors in each said computing module, if no fault is detected in the system including each said computing module, generating an interruption to all of said processors, and

causing each said processor to execute a synchronization control instruction to adjust timing of a response to an access from each processor, thereby causing each computing module to resume operation in synchronization.

7. The re-synchronization method as set forth in claim 6, further comprising the steps of:

detecting existence/non-existence of a fault in the entire system including each said computing module,

monitoring an access of the processor in each said computing module to the external bus,

when detecting disagreement in output among the respective computing modules, if no fault is detected in the system, notifying an interruption to each said processor, and

causing each said processor to execute the clock synchronization control instruction to adjust timing of a response to an access from each processor, thereby causing each computing module to resume operation in synchronization.

8. The re-synchronization method as set forth in claim 7, further comprising the steps of:

when detecting disagreement in output among the respective computing modules, if no fault is detected in the system, interrupts each said processor with a predetermined task for re-synchronizing the respective computing modules which is a task of executing an access to a predetermined resource;

queuing access to said resource from each processor, and

responding to said accesses from all the computing modules simultaneously when all the accesses from said processors are received.

9. A re-synchronization program for executing re-

synchronization processing of a lock-step synchronism fault-tolerant computer system including a plurality of computing modules having a processor and a memory in which each computing module processes the same instruction string in synchronization with each other, comprising the functions of:

when detecting disagreement in a state of access to an external bus among respective said processors in each said computing module, if no fault is detected in the system including each said computing module, generating an interruption to all of said processors, and

causing each said processor to execute a clock synchronization control instruction to adjust timing of a response to an access from each processor, thereby causing each computing module to resume operation in synchronization.

10. The re-synchronization program as set forth in claim 9, further comprising the functions of:

detecting existence/non-existence of a fault in the entire system including each said computing module, monitoring an access of the processor in each said computing module to the external bus,

when detecting disagreement in output among the respective computing modules, if no fault is detected in the system, notifying an interruption to each said

processor, and

causing each said processor to execute the synchronization control instruction to adjust timing of a response to an access from each processor, thereby causing each computing module to resume operation in synchronization.

11. The re-synchronization program as set forth in claim 10, further comprising the functions of:

when detecting disagreement in output among the respective computing modules, if no fault is detected in the system, interrupts each said processor with a predetermined task for re-synchronizing the respective computing modules which is a task of executing an access to a predetermined resource;

queuing access to said resource from each processor, and

responding to said accesses from all the computing modules simultaneously when all the accesses from said processors are received.